

CGRateS-OpenSIPS Integration

Dan Christian Bogos
dan.bogos@itsyscom.com

OpenSIPS Summit, Amsterdam 2017



Our Background



Located in Bavaria/Germany, over 10 years of experience with architecting server side solutions in VoIP environment

Platform implementations covering both wholesale and retail business categories

Responsibly understanding real-time processing constraints and the seriousness of live system outages

About CGRateS

Real-time Enterprise Billing Suite

Pluggable into existing infrastructure

Accommodate new components into ISP/ITSP network (eg: new VoIP switch, SMS Service)

Non-intrusive into existing setups

Open Source software

Full sources available on Github repository

No add-ons in private repositories

Strong and vibrant community behind

Performance Oriented

Built-in advanced cache system (transactional, LRU + TTL records)

Asynchronous processing with micro-threads

Test driven development

More than. 1500 tests as part of the build system

About CGRateS (2)

Modular architecture

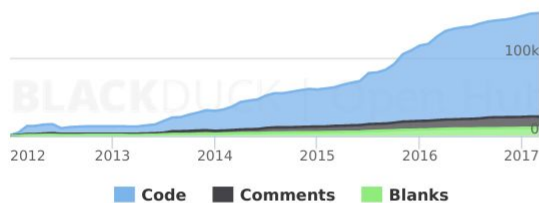
Cloud-ready, micro-services with rich set of RPC APIs
Easy to enhance by rewriting specific components

Feature-rich

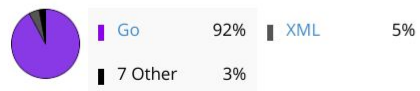
Online/Offline Charging System (OCS)
Multi-tenancy from day one
Rating Engine with Derived Charging and A-Number rating
Account Balances Management with Bundles
Session or Event Charging with balance reservation and refunds
CDR logging with support for Interim Records
Fraud detection with automatic mitigation
LCR with QoS/Bundles
Call Statistics with pattern monitoring
Diameter/Radius Server with process templates (standard agnostic)
Resource allocation controller
Built-in High-Availability support
Agile in developing new features

Code

Lines of Code



Languages



Activity

Commits per Month



30 Day Summary

Feb 8 2017 — Mar 10 2017

74 Commits

7 Contributors including 4 new contributors

12 Month Summary

Mar 10 2016 — Mar 10 2017

1212 Commits

Down -456 (27%) from previous 12 months

26 Contributors

Up + 14 (116%) from previous 12 months

Community

Contributors per Month



Most Recent Contributors

- danbogos
- ...hristian Bogos
- Edwardro22
- alin104n
- Wasim Baig
- mrgab

Ratings

Be the first to rate this project

Click to add your rating



Review this Project!

Actively maintained

*stats provided by openhub.net



<p>RATE/ACCOUNT/LCR</p> <p>Functionality:</p> <ul style="list-style-type: none"> • calculate costs for events • maintain accounts • compute LCR • real-time fraud mitigation 	<p>CDR SERVER</p> <p>Functionality:</p> <ul style="list-style-type: none"> • centralized CDR server • CDR replication • forward to CDRStats 	<p>SESSION MANAGER</p> <p>Functionality:</p> <ul style="list-style-type: none"> • maintain/disconnect sessions • balance reservation • balance refunds
<p>DIAMETER AGENT</p> <p>Functionality:</p> <ul style="list-style-type: none"> • call control via diameter interface (rfc 4006). 	<p>CDR STATS</p> <p>Functionality:</p> <ul style="list-style-type: none"> • compute real-time CDR stats • real-time fraud mitigation 	<p>RESOURCE LIMITER</p> <p>Functionality:</p> <ul style="list-style-type: none"> • control resource allocation
<p>USER SERVER</p> <p>Functionality:</p> <ul style="list-style-type: none"> • maintain user profiles (LDAP similarity) 	<p>ALIASING SERVER</p> <p>Functionality:</p> <ul style="list-style-type: none"> • alias request/reply information using predefined rules 	<p>PUBSUB SERVER</p> <p>Functionality:</p> <ul style="list-style-type: none"> • expose internal events to subscribed external components

CGRateS popular Subsystems

CGR-RALs (Rating/Accounting)

Highly configurable rating

Connect fees, rate units, rate increments, rates grouping, a-number rating, various rounding methods, configurable decimals in costs, maximum cost per destination with hit strategy, rating profile scheduling

Unlimited Balances per Account

*voice, *data, *sms, *mms, *monetary, *generic
Unlimited bundle combinations with balance prioritisation

Concurrent sessions handling

Balance reservation in chunks of debit interval
Balance refunds
Debit sleep when needed

CGR-RALs (LCR)

Core component logic

Internally or remotely accessible through APIer or RALs components
Non-intrusive, injects supplier information into Telecom Switch

Tightly coupled with ACCOUNTING subsystem

Provides LCR over bundles

Integrates traffic patterns

Computes LCR for specific call duration

Extended functionality through multiple strategies

*static, *least_cost, *highest_cost, *qos_thresholds, *qos, *load_distribution
Flexible strategy parameters

CDR SERVER

Real-time CDR Server

Accessible Internal, GOB, JSON, HTTP-JSON, HTTP-REST interfaces

Offline CDR Import (csv, xml, fwv)

Automated via Linux inotify or scheduled

Simultaneous folders monitored with multiple import templates per folder

Dynamic fields via import templates

Zero configuration CDR Sources

Asterisk

FreeSWITCH

Kamailio

OpenSIPS

CDR SERVER (2)

Derived Charging support

Real-time CDR replication

Raw or Rated CDRs

CDR Exporter

CSV, Fixed Length Fields, Combined
Export templates

CDR STATS

Standalone component

Internally or remotely accessible
Performance oriented

RawCDR and RatedCDR sources

Multiple Stats Queues

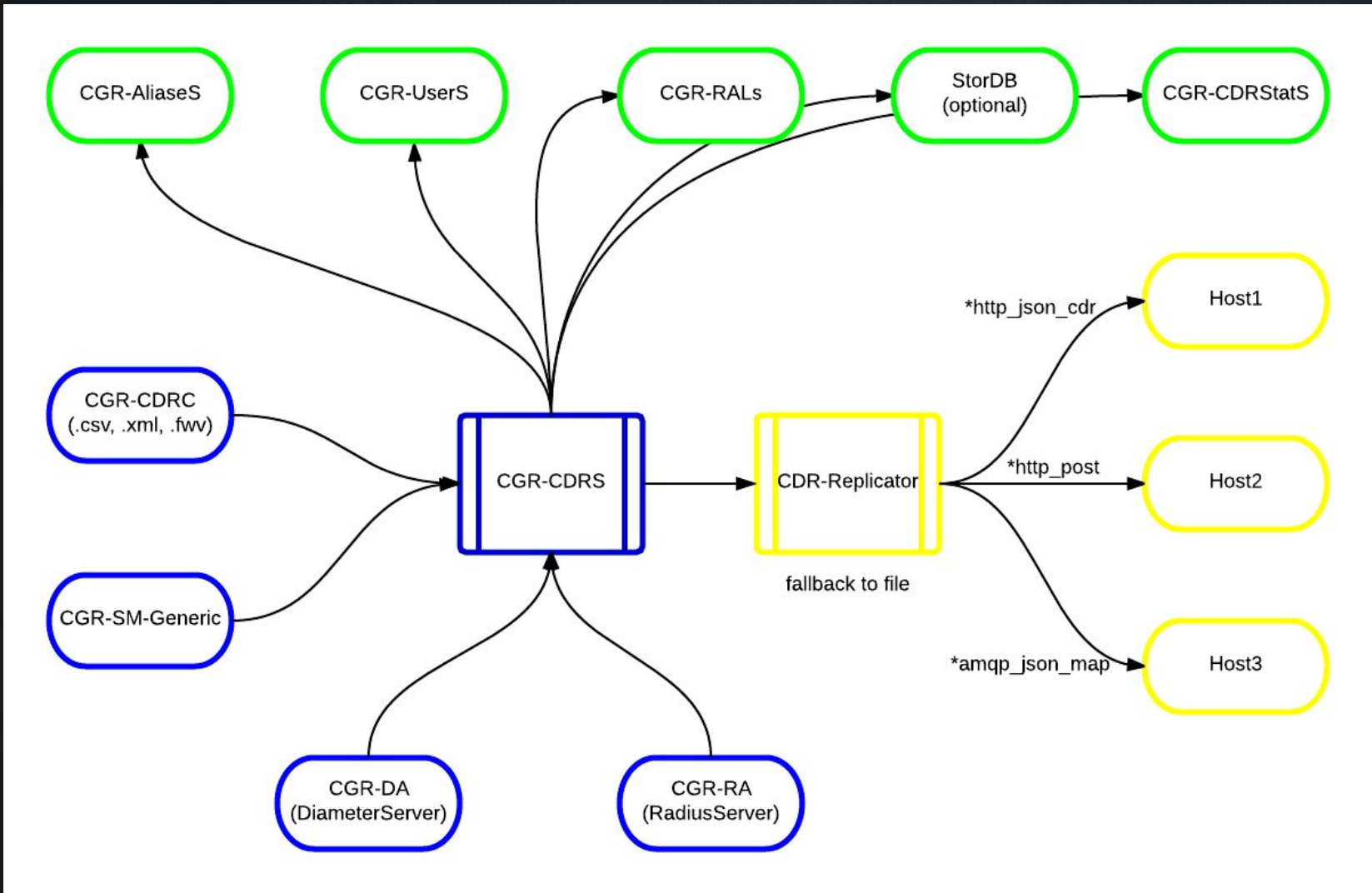
Individually stat queues for same CDR

Highly configurable Stats Queues

QueueLength, TimeWindow, Metrics
CDR Field Filters
Multiple metrics: ASR, ACD, ACC, TCC, DCC, PDD,

ActionTriggers attached to each queue

One-time, recurrent triggers
Synchronous & Asynchronous Actions executed



CGRateS CDR Processing logic

FRAUD MITIGATION

Part of Accounting

Tightly integrated, balance operations cannot avoid it
Minimum & maximum balance/counter monitors

Part of CDRStatS

Multiple metrics and stat queues thresholds

Part of RLs

Monitoring usage counters

Synchronous & Asynchronous Actions

OpenSIPS Integration

Prior to OpenSIPS 2.3 (old way)

rest_client for authorization and accounting
mi_datagram for call disconnects coming from CGRateS side
db_flatstore/ radius for CDRs

After OpenSIPS 2.3 (new way)

Native CGRateS module for everything ;)
Thanks to OpenSIPS core developers!

OpenSIPS scripting

Module configuration

```
loadmodule "cgrates.so"  
modparam("cgrates", "cgrates_engine", "127.0.0.1:2012")
```

Call authorization

```
...  
    $cgr(Tenant) = $fd;  
    $cgr(RequestType) = "*prepaid";  
    if (!async(cgrates_auth("$fU", "$rU"), resume_cgr_auth)) {  
        sl_send_reply("503", "Service Unavailable");  
        Exit;  
    }  
  
...  
route [resume_cgr_auth] {  
    if ($rc < 0) {  
        xlog("Call not authorized: code=$cgrret!\n");  
        send_reply("403", "Forbidden");  
        exit;  
    }  
  
...  
...
```

OpenSIPS scripting (2)

Call accounting

```
cgrates_acc("cdr|missed", "$fU", "$rU");
```

Generic RPC Call

```
...  
$cgr(Tenant) = $fd;  
$cgr(Account) = $fU;  
$cgr(OriginID) = $ci;  
$cgr(SetupTime) = "" + $Ts;  
$cgr(RequestType) = "*prepaid";  
$cgr(Destination) = $rU;  
cgrates_cmd("SMGenericV1.GetMaxUsage");  
xlog("Call is allowed to run $cgrret seconds\n");  
...
```

Troubleshooting

OpenSIPS logs

```
xlog("Call is allowed to run $cgrret seconds\n");
```

Protocol capture

```
T 2017/04/25 20:49:48.601645 127.0.0.1:37245 -> 127.0.0.1:2012 [AP]
{ "method": "SMGenericV1.MaxUsage", "params": [ { "OriginID":
"f6d44d965477901c8bda362990232973@0:0:0:0:0:0:0:0", "Account": "1001", "SetupTime":
"1493146188", "Destination": "1002" } ] }
##
T 2017/04/25 20:49:48.626356 127.0.0.1:2012 -> 127.0.0.1:37245 [AP]
{"id":null,"result":10800,"error":null}
```

Syslog analysis

```
Apr 27 20:35:31 CGRDev1 CGRates <8de1d169-a2b3-4fa7-a213-c82c7eaa169f> [7128]: Starting
CGRates CDRS service.
Apr 27 20:35:31 CGRDev1 CGRates <8de1d169-a2b3-4fa7-a213-c82c7eaa169f> [7128]: <CDRC> No
enabled CDRC clients
Apr 27 20:35:31 CGRDev1 CGRates <8de1d169-a2b3-4fa7-a213-c82c7eaa169f> [7128]: Starting
CGRates SMGeneric service.
```

Questions?

Website

<http://www.cgrates.org>

Documentation

<http://cgrates.readthedocs.org>

Code + issues tracker

<https://github.com/cgrates/cgrates>

Support

Google group: **CGRateS**

IRC Freenode: **#cgrates**